AMENDMENTS TO THE CLAIMS:

Claim 1. (Currently amended) A method for storing information in a recoverable manner on an untrusted system, comprising:

sending, by a client, a request to a recovery server for recovery of a failed database;

determining whether said request is legitimate;

based on said determining, sending an old local key to the client;

decrypting by said client the failed database with the old local key, to recover the failed database; and

re-encrypting the recovered database with a new local key,

wherein at least one of said old local key and said new local key is based upon at least one unique characteristic of a hardware component associated with said database.

Claim 2. (Original) The method of claim 1, further comprising:

verifying whether a key database identification has been tampered with.

Claim 3. (Original) The method of claim 1, wherein said database is associated with content which is purchased from a content owner and stored, along with a keyword or codeword, on the database of the client.

Claim 4. (Original) The method of claim 3, wherein the client can access the recovery server with the keyword to restore the database.

Claim 5. (Currently amended) The method of claim 1, wherein said at least one of said old local key and said new local key comprises one of a unique key ~~is provided~~ for each piece of content in said database and an overall key ~~is provided~~ for the entire database.

Claim 6. (Canceled).

Claim 7. (Currently amended) The method of claim 1, wherein at least one of said old local key and said new local key is ~~the keys are~~ based on at least one of a processor identification, a particular sector of a system file and random data stored in a non-volatile area of a computer system of said client.

Claim 8. (Currently amended) The method of claim 7, wherein said random data comprises ~~includes~~ values placed in a secret location ~~in the system, said secret location~~ comprising ~~including~~ any of a system's basic input/output system (BIOS), a nonvolatile RAM (NVRAM), and a hard disk.

Claim 9. (Currently amended) The method of claim 1, wherein said at least one of said old local key and said new ~~the~~ local key is further based ~~keys are dependent~~ on a value in at least one secret location which changes every time a predetermined action occurs.

Claim 10. (Currently amended) The method of claim 9, further comprising:

storing a counter in the secret, nonvolatile location; and

incrementing the counter ~~incremented~~.

Claim 11. (Currently amended) The method of claim 9, further comprising:

incrementing a counter periodically; and

storing the counter ~~stored~~ to the nonvolatile location; such that a restored value will be saved with a wrong key.

Claim 12. (Currently amended) The method of claim 1, wherein said local ~~a unique~~ key is based on ~~produced by using~~ a combination of values stored in a local storage and a nonvolatile location of a computer system of said client.

Claim 13. (Currently amended) The method of claim 1, wherein the database is encrypted with said ~~a~~ local key ~~that is used with the database~~, and ~~the~~ said local key is ~~encrypted such that it~~ ~~is~~ decryptable only by the recovery server.

Claim 14. (Currently amended) The method of claim 13, further comprising decrypting the old local key at ~~wherein~~ the recovery server using ~~uses~~ public key cryptography ~~to decrypt the local key~~.

Claim 15. (Currently amended) The method of claim 1, wherein said recovery server

automatically provides said old local key in response to ~~at~~ a first request ~~thereof~~.

Claim 16. (Currently amended) The method of claim 15, wherein said request is sent from the client to the recovery server if the old ~~new~~ local key is not correct, wherein said request further comprises ~~the client extracts~~ the encrypted old local key~~, and sends it to the recovery server and the recovery server judges whether to allow the recovery~~.

Claim 17. (Currently amended) The method of claim 16, wherein if the recovery server determines that said request is legitimate said method further comprises ~~comprising~~:

decrypting the old local key ~~and sending it back to the client for decrypting the information and re-encrypting the decrypted information with a new valid local key~~.

Claim 18. (Currently amended) The method of claim 1, wherein ~~the~~ data is stored in a non-volatile area of a machine of said client, and further comprising:

changing said data and said old local key ~~is changed~~ every time a count changes~~, such that said local key also changes~~.

Claim 19. (Currently amended) The method of claim 1, wherein re-encrypting the recovered database with the new local key comprises:

encrypting a random key using said new local key; and

re-encrypting the recovered database using ~~wherein~~ said random key ~~keys are used to~~

~~encrypt the data, and the local key is used to encrypt the random keys~~.

Claim 20. (Original) The method of claim 1, wherein counters are kept in records of the database, and the local key is used to encrypt the counters.

Claim 21. (Currently amended) The method of claim 1, wherein the request comprises ~~includes~~ a header and a body, and wherein all but a first portion of the header is ~~being~~ encrypted with said old ~~a~~ local ~~encryption~~ key.

Claim 22. (Currently amended) The method of claim 21, wherein said header comprises a cleartext portion of the header that comprises ~~contains~~ a unique database identfication ~~ID which is unique among all the users, said ID serving as an identification during recovery~~.

Claim 23. (Currently amended) The method of claim 22, wherein said header further comprises a second portion comprising ~~of the header is a combined item which contains the~~ said old local ~~encryption~~ key and the database identification ~~ID~~, wherein said second portion of said header is encrypted with a public key from the recovery server ~~center's public key, such that this item is only in the clear regarding the local key database key~~.

Claim 24. (Currently amended) The method of claim 23, wherein said failed database comprises said header and ~~a remaining portion of the key database includes~~ fields which are

encrypted with the old local key~~, including the rest of the header, which includes the key database ID and~~ wherein said header further comprises a codeword~~, said fields serving to check whether the key is calculated correctly or the system has been modified or tampered with~~.

Claim 25. (Currently amended) The method of claim 24, wherein said failed database comprises a body ~~an entirety of said body is~~ encrypted with the old local ~~encryption~~ key.

Claim 26. (Currently amended) The method of claim 25, wherein said old local ~~a decryption~~ key for the database cannot be reconstructed locally and must explicitly be recovered, such that a client application program extracts said ~~the~~ second portion ~~field~~ from said ~~a~~ cleartext portion of the ~~key database~~ header,

wherein said second portion comprises said old local ~~field containing a lost~~ key that ~~and~~ is encrypted with said ~~the recovery center's~~ public key~~, so that only the recovery server can decrypt it using its private key~~; wherein sending said request from said client comprises sending the second portion ~~field~~ to the recovery server~~, for checking the legality of this action, decrypting the key and returning the key so that a client application decrypts the old key database, generates a new key and sets the system parameters and encrypts the key database~~.

Claim 27. (Currently amended) The method of claim 1, wherein said determining whether said request is legitimate comprises:

determining ~~the recovery server determines whether or not to automatically grant the~~

~~recover operation~~ based on any of whether a normal user upgrade is due, and whether a predetermined time period has elapsed between a user recovery of a failing machine.

Claim 28. (Currently amended) The method of claim 1, wherein said determining whether said request is legitimate comprises:

~~operator-assisted recovery is performed by~~ resetting ~~certain~~ parameters in a ~~the~~ decision logic; and

requesting another request from ~~making~~ the client ~~re-request recovery~~.

Claim 29. (Currently amended) A method of allowing recovery of a proprietary database, comprising:

~~detecting, by a recovery server,~~ receiving, from a client at a recovery server, a request to restore a database;

determining, by the recovery server, whether the request is legitimate by verifying ~~an identification (ID) of~~ a key database identification included in the request of the user;

if the key database identification matches a predetermined identification ~~based on the ID matching a predetermined ID~~, then applying a recovery decision logic, and granting the restore request to the client by the recovery server;

forwarding an old ~~a~~ local key ~~that the database was incorporated with~~ to a user; and

~~using the local key,~~ calculating a new local key by decrypting the database with said old ~~the~~ local key by said client, ~~such that the database is re-encrypted with the new local key~~

wherein at least one of said old local key and said new local key is based upon at least one unique characteristic of a hardware component associated with said database.

Claim 30. (Currently amended) The method of claim 29, further comprising:

~~wherein operator-assisted recovery is performed by~~ resetting certain parameters in the decision logic; and

requesting another request from ~~making~~ the client ~~re-request recovery~~.

Claim 31. (Currently amended) A system for storing information in a recoverable manner on an untrusted system, comprising:

means for sending, by a client, a request to a recovery server for recovery of a failed database;

means for determining whether said request is legitimate;

based on an output from said means for determining, means for sending an old local key to the client;

means for decrypting, by said client, the failed database with the old local key, ~~to recover the failed database~~; and

means for re-encrypting the recovered database with a new local key.

wherein at least one of said old local key and said new local key is based upon at least one unique characteristic of a hardware component associated with said database.

Claim 32. (Currently amended) The system of claim 31, further comprising:

~~wherein operator-assisted recovery is performed by~~ means for resetting ~~certain~~ parameters in a ~~the~~ decision logic; and

means for requesting another request from ~~making~~ the client ~~re-request recovery~~.

Claim 33. (Currently amended) A system of allowing recovery of a proprietary database, comprising:

means for receiving ~~detecting~~, by a recovery server, a request from a client to restore a database;

means for determining~~, by the recovery server,~~ whether the request is legitimate by verifying ~~an identification (ID) of~~ a key database identification included in the request of the client ~~user~~;

means for applying a recovery decision logic based on the key database identification ~~ID~~ matching a predetermined identification ~~ID~~, and for granting the restore request to the client by the recovery server;

means for forwarding an old local key ~~that the database was incorporated with~~ to said client ~~a user~~;

means for~~, using the old local key, for calculating a new local key by~~ decrypting the

database with the old local key, and calculating a ~~such that the database is re-encrypted with the~~ new local key,

wherein at least one of said old local key and said new local key is based upon at least one unique characteristic of a hardware component associated with said database.

Claim 34. (Currently amended) The system of claim 33, further comprising

~~wherein operator-assisted recovery is performed by~~ means for resetting ~~certain~~ parameters in a ~~the~~ decision logic; and

means for requesting another request from ~~making~~ the client ~~re-request recovery~~.

Claim 35. (Currently amended) A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of storing information in a recoverable manner on an untrusted system, comprising:

sending, by a client, a request to a recovery server for recovery of a failed database;

determining whether said request is legitimate;

based on said determining, sending a local key to the client;

decrypting by said client the failed database with the local key~~, to recover the failed database~~; and

re-encrypting the ~~recovered~~ decrypted database with a new key,

wherein at least one of said local key and said new key is based upon at least one unique characteristic of a hardware component associated with said database.

Claim 36. (Currently amended) A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of allowing recovery of a proprietary database, comprising:

receiving a restore request from a client ~~detecting~~, by a recovery server~~, a request to restore a database~~;

determining, by the recovery server, whether the request is legitimate by verifying ~~an identification (ID) of~~ a key database identification included in the request of the client ~~user~~;

based on the key database identification ~~ID~~ matching a predetermined identification ~~ID~~, ~~then~~ applying a recovery decision logic, and granting the restore request by the recovery server;

forwarding an old ~~a~~ local key from said recovery server ~~that the database was incorporated with~~ to said client ~~a user~~;

decrypting the database using the old local key; and

~~,~~calculating a new local key ~~by decrypting the database with the local key, such that the database is re-encrypted with the new local key~~,

wherein at least one of said old local key and said new local key is based upon at least one unique characteristic of a hardware component associated with said database.

Claim 37. (New) The method of claim 7, wherein said non-volatile area of said computer system comprises an area of a computer system that is not protected by a backup operation.